LISP Traffic Engineering Use-Cases
draft-farinacci-lisp-te-03

Abstract

   This document describes how LISP reencapsulating tunnels can be used
   for Traffic Engineering purposes. The mechanisms described in this
   document require no LISP protocol changes but do introduce a new
   locator (RLOC) encoding. The Traffic Engineering features provided
   by these LISP mechanisms can span intra-domain, inter-domain, or
   combination of both.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on January 11, 2014.

Table of Contents

1. Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [RFC2119].

[2](#). **Introduction**

   This document describes the Locator/Identifier Separation Protocol
   (LISP), which provides a set of functions for routers to exchange
   information used to map from non globally routeable Endpoint
   Identifiers (EIDs) to routeable Routing Locators (RLOCs). It also
   defines a mechanism for these LISP routers to encapsulate IP packets
   addressed with EIDs for transmission across the Internet that uses
   RLOCs for routing and forwarding.

   When LISP routers encapsulate packets to other LISP routers, the path
   stretch is typically 1, meaning the packet travels on a direct path
   from the encapsulating ITR to the decapsulating ETR at the
   destination site. The direct path is determined by the underlying
   routing protocol and metrics it uses to find the shortest path.

   This specification will examine how reencapsulating tunnels [RFC6830]
   can be used so a packet can take an adminstratively specified path, a
   congestion avoidance path, a failure recovery path, or multiple load-
   shared paths, as it travels from ITR to ETR. By introducing an
   Explicit Locator Path (ELP) locator encoding [LISP-LCAF], an ITR can
   encapsulate a packet to a Reencapsulating Tunnel Router (RTR) which
   decapsulates the packet, then encapsulates it to the next locator in
   the ELP.

3. Definition of Terms

   Endpoint ID (EID):   An EID is a 32-bit (for IPv4) or 128-bit (for
      IPv6) value used in the source and destination address fields of
      the first (most inner) LISP header of a packet. The host obtains
      a destination EID the same way it obtains an destination address
      today, for example through a Domain Name System (DNS) [RFC1034]
      lookup or Session Invitation Protocol (SIP) [RFC3261] exchange.
      The source EID is obtained via existing mechanisms used to set a
      host's "local" IP address. An EID used on the public Internet
      must have the same properties as any other IP address used in that
      manner; this means, among other things, that it must be globally
      unique. An EID is allocated to a host from an EID-prefix block
      associated with the site where the host is located. An EID can be
      used by a host to refer to other hosts. EIDs MUST NOT be used as
      LISP RLOCs. Note that EID blocks MAY be assigned in a
      hierarchical manner, independent of the network topology, to
      facilitate scaling of the mapping database. In addition, an EID
      block assigned to a site may have site-local structure
      (subnetting) for routing within the site; this structure is not
      visible to the global routing system. In theory, the bit string
      that represents an EID for one device can represent an RLOC for a
      different device. As the architecture is realized, if a given bit
      string is both an RLOC and an EID, it must refer to the same
      entity in both cases. When used in discussions with other
      Locator/ID separation proposals, a LISP EID will be called a
      "LEID". Throughout this document, any references to "EID" refers
      to an LEID.

   Routing Locator (RLOC):   A RLOC is an IPv4 [RFC0791] or IPv6
      [RFC2460] address of an egress tunnel router (ETR). A RLOC is the
      output of an EID-to-RLOC mapping lookup. An EID maps to one or
      more RLOCs. Typically, RLOCs are numbered from topologically-
      aggregatable blocks that are assigned to a site at each point to
      which it attaches to the global Internet; where the topology is
      defined by the connectivity of provider networks, RLOCs can be
      thought of as PA addresses. Multiple RLOCs can be assigned to the
      same ETR device or to multiple ETR devices at a site.

   Reencapsulating Tunnel Router (RTR):   An RTR is a router that acts
      as an ETR (or PETR) by decapsulating packets where the destination
      address in the "outer" IP header is one of its own RLOCs. Then
      acts as an ITR (or PITR) by making a decision where to encapsulate
      the packet based on the next locator in the ELP towards the final
      destination ETR.

   Explicit Locator Path (ELP):   The ELP is an explicit list of RLOCs
      for each RTR a packet must travel to along its path toward a final
      destination ETR (or PETR). The list is a strict ordering where
      each RLOC in the list is visited. However, the path from one RTR
      to another is determined by the underlying routing protocol and
      how the infrastructure assigns metrics and policies for the path.

   Recursive Tunneling: Recursive tunneling occurs when a packet has
      more than one LISP IP header. Additional layers of tunneling MAY
      be employed to implement traffic engineering or other re-routing
      as needed. When this is done, an additional "outer" LISP header
      is added and the original RLOCs are preserved in the "inner"
      header. Any references to tunnels in this specification refers to
      dynamic encapsulating tunnels and they are never statically
      configured.

   Reencapsulating Tunnels: Reencapsulating tunneling occurs when an
      ETR removes a LISP header, then acts as an ITR to prepend another
      LISP header. Doing this allows a packet to be re-routed by the
      reencapsulating router without adding the overhead of additional
      tunnel headers. Any references to tunnels in this specification
      refers to dynamic encapsulating tunnels and they are never
      statically configured. When using multiple mapping database
      systems, care must be taken to not create reencapsulation loops
      through misconfiguration.

[4](#). **Overview**

   Typically, a packet's path from source EID to destination EID travels
   through the locator core via the encapsulating ITR directly to the
   decapsulating ETR as the following diagram illustrates:

   Legend:

   seid: Packet is originated by source EID 'seid'.

   deid: Packet is consumed by destination EID 'deid'.

   A,B,C,D : Core routers in different ASes.

   ---> : The physical topological path between two routers.

   ===> : A multi-hop LISP dynamic tunnel between LISP routers.


```
                             Core Network
   Source site       (----------------------------)    Destination Site
   +--------+         (                            )          +---------+
   |        \         (                            )         /          |
   | seid     ITR ---(---> A --> B --> C --> D ---)---> ETR      deid |
   |        / ||      (                            )     ^^ \          |
   +--------+  ||     (                            )     ||  +---------+
              ||    (--------------------------)     ||
              ||                                      ||
              ||                                      ||
              ========================================
                             LISP Tunnel
```
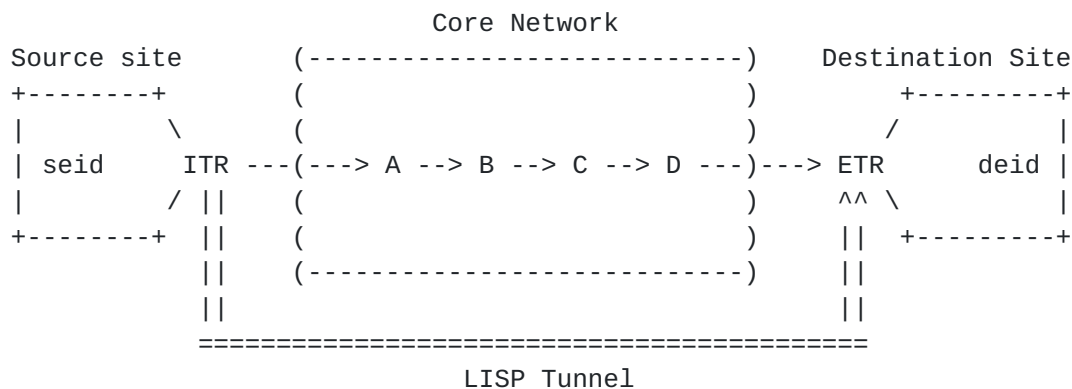

                 Typical Data Path from ITR to ETR

   Let's introduce RTRs 'X' and 'Y' so that, for example, if it is
   desirable to route around the path from B to C, one could provide an
   ELP of (X,Y,etr):

```
                         Core Network
  Source site       (----------------------------)    Destination Site
  +--------+        (                            )         +---------+
  |        \        (                            )        /          |
  | seid     ITR ---(---> A --> B --> C --> D ---)---> ETR      deid |
  |        / ||     (           /       ^        )    ^^ \           |
  |       /  ||     (           |        \       )    || \          |
  +-------+  ||     (           v         |      )    ||  +--------+
             ||     (          X ======> Y       )    ||
             ||     (          ^^         ||      )    ||
             ||     (--------||---------||-------)    ||
             ||             ||         ||            ||
          =================         =================
             LISP Tunnel               LISP Tunnel
```
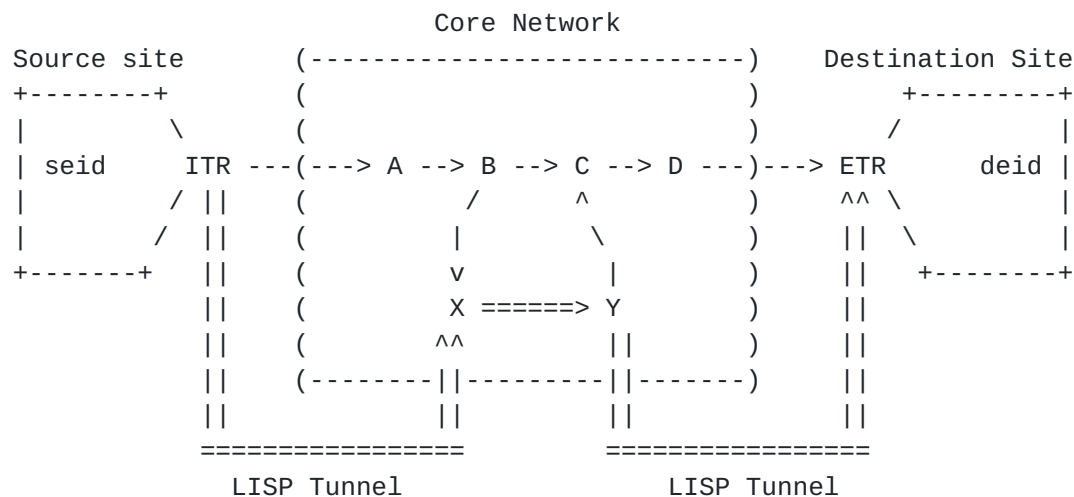
      ELP tunnel path ITR ==> X, then X ==> Y, and then Y ==> ETR

   There are various reasons why the path from 'seid' to 'deid' may want
   to avoid the path from B to C. To list a few:

   o There may not be sufficient capacity provided by the networks that
      connect B and C together.

   o There may be a policy reason to avoid the ASes that make up the
      path between B and C.

   o There may be a failure on the path between B and C which makes the
      path unreliable.

   o There may be monitoring or traffic inspection resources close to
      RTRs X and Y that do network accounting or measurement.

   o There may be a chain of services performed at RTRs X and Y
      regardless if the path from ITR to ETR is through B and C.

[5](#). **Explicit Locator Paths**

   The notation for a general formatted ELP is (x, y, etr) which
   represents the list of RTRs a packet SHOULD travel through to reach
   the final tunnel hop to the ETR.

   The procedure for using an ELP at each tunnel hop is as follows:

   1. The ITR will retrieve the ELP from the mapping database.

   2. The ITR will encapsulate the packet to RLOC 'x'.

   3. The RTR with RLOC 'x' will decapsulate the packet. It will use
       the decapsulated packet's destination address as a lookup into
       the mapping database to retrieve the ELP.

   4. RTR 'x' will encapsulate the packet to RTR with RLOC 'y'.

   5. The RTR with RLOC 'y' will decapsulate the packet. It will use
       the decapsulated packet's destination address as a lookup into
       the mapping database to retrieve the ELP.

   6. RTR 'y' will encapsulate the packet on the final tunnel hop to
       ETR with RLOC 'etr'.

   7. The ETR will decapsulate the packet and deliver the packet to the
       EID inside of its site.

   The specific format for the ELP can be found in [[LISP-LCAF](#)]. It is
   defined that an ELP will appear as a single encoded locator in a
   locator-set. Say for instance, we have a mapping entry for EID-
   prefix 10.0.0.0/8 that is reachable via 4 locators. Two locators are
   being used as active/active and the other two are used as active/
   active if the first two go unreachable (as noted by the priority
   assignments below). This is what the mapping entry would look like:

   EID-prefix: 10.0.0.0/8
   Locator-set: ETR-A: priority 1, weight 50
                 ETR-B: priority 1, weight 50
                 ETR-C: priority 2, weight 50
                 ETR-D: priority 2, weight 50

   If an ELP is going to be used to have a policy path to ETR-A and
   possibly another policy path to ETR-B, the locator-set would be
   encoded as follows:

```
   EID-prefix:   10.0.0.0/8
   Locator-set:  (x, y, ETR-A): priority 1, weight 50
                 (q, r, ETR-B): priority 1, weight 50
                 ETR-C:         priority 2, weight 50
                 ETR-D:         priority 2, weight 50
```

The mapping entry with ELP locators is registered to the mapping database system just like any other mapping entry would. The registration is typically performed by the ETR(s) that are assigned and own the EID-prefix. That is, the destination site makes the choice of the RTRs in the ELP. However, it may be common practice for a provisioning system to program the mapping database with ELPs.

Another case where a locator-set can be used for flow-based load-sharing across multiple paths to the same destination site:

```
   EID-prefix:   10.0.0.0/8
   Locator-set:  (x, y, ETR-A): priority 1, weight 75
                 (q, r, ETR-A): priority 1, weight 25
```

Using this mapping entry, an ITR would load split 75% of the EID flows on the (x, y, ETR-A) ELP path and 25% of the EID flows on the (q, r, ETR-A) ELP path. If any of the ELPs go down, then the other can take 100% of the load.

5.1. ELP Re-optimization

ELP re-optimization is a process of changing the RLOCs of an ELP due to underlying network change conditions. Just like when there is any locator change for a locator-set, the procedures from the main LISP specification [RFC6830] are followed.

When a RLOC from an ELP is changed, Map-Notify messages [RFC6833] can be used to inform the existing RTRs in the ELP so they can do a lookup to obtain the latest version of the ELP. Map-Notify messages can also be sent to new RTRs in an ELP so they can get the ELP in advance to receiving packets that will use the ELP. This can minimize packet loss during mapping database lookups in RTRs.

5.2. Using Recursion

In the previous examples, we showed how an ITR encapsulates using an ELP of (x, y, etr). When a packet is encapsulated from the ITR to RTR 'x', the RTR may want a policy path to RTR 'y' and run another level of reencapsulating tunnels for packets destined to RTR 'y'. In this case, RTR 'x' does not decapsulate packets from the ITR, but rather performs a mapping database lookup on the address 'y'. This

   can be done when using a public or private mapping database. The
   decision to use address 'y' as an encapsulation address versus a
   lookup address is based on the L-bit setting for 'y' in the ELP
   entry. The decision and policy of ELP encodings are local to the
   entity which registers the EID-prefix associated with the ELP.

   Another example of recursion is when the ITR uses the ELP (x, y, etr)
   to first prepend a header with a destination RLOC of the ETR and then
   prepend another header and encapsulate the packet to RTR 'x'. When
   RTR 'x' decapsulates the packet, rather than doing a mapping database
   lookup on RTR 'y' the last example showed, instead RTR 'x' does a
   mapping database lookup on ETR 'etr'. In this scenario, RTR 'x' can
   choose an ELP from the locator-set by considering the source RLOC
   address of the ITR versus considering the source EID.

   This additional level of recursion also brings advantages for the
   provider of RTR 'x' to store less state. Since RTR 'x' does not need
   to look at the inner most header, it does not need to store EID
   state. It only stores an entry for RTR 'y' which many EID flows
   could share for scaling benefits. The locator-set for entry 'y'
   could either be a list of typical locators, a list of ELPs, or
   combination of both. Another advantage is that packet load-splitting
   can be accomplished by examining the source of a packet. If the
   source is an ITR versus the source being the last-hop of an ELP the
   last-hop selected, different forwarding paths can be used.

5.3. ELP Selection based on Class of Service

   Paths to an ETR may want to be selected based on different classes of
   service. Packets from a set of sources that have premium service can
   use ELP paths that are less congested where normal sources use ELP
   paths that compete for less resources or use longer paths for best
   effort service.

   Using source/destination lookups into the mapping database can yield
   different ELPs. So for example, a premium service flow with
   (source=1.1.1.1, dest=10.1.1.1) can be described by using the
   following mapping entry:

   EID-prefix: (1.0.0.0/8, 10.0.0.0/8)
   Locator-set: (x, y, ETR-A): priority 1, weight 50
                 (q, r, ETR-A): priority 1, weight 50

   And all other best-effort sources would use different mapping entry
   described by:

```
   EID-prefix:   (0.0.0.0/0, 10.0.0.0/8)
   Locator-set:  (x, x', y, y', ETR-A): priority 1, weight 50
                 (q, q', r, r', ETR-A): priority 1, weight 50
```

   If the source/destination lookup is coupled with recursive lookups,
   then an ITR can encapsulate to the ETR, prepending a header that
   selects source address ITR-1 based on the premium class of service
   source, or selects source address ITR-2 for best-effort sources with
   normal class of service. The ITR then does another lookup in the
   mapping database on the prepended header using lookup key
   (source=ITR-1, dest=10.1.1.1) that returns the following mapping
   entry:

```
   EID-prefix:   (ITR-1, 10.0.0.0/8)
   Locator-set:  (x, y, ETR-A): priority 1, weight 50
                 (q, r, ETR-A): priority 1, weight 50
```

   And all other sources would use different mapping entry with a lookup
   key of (source=ITR-2, dest=10.1.1.1):

```
   EID-prefix:   (ITR-2, 10.0.0.0/8)
   Locator-set:  (x, x', y, y', ETR-A): priority 1, weight 50
                 (q, q', r, r', ETR-A): priority 1, weight 50
```

   This will scale the mapping system better by having fewer source/
   destination combinations. Refer to the Source/Dest LCAF type
   described in [LISP-LCAF] for encoding EIDs in Map-Request and Map-
   Register messages.

5.4. Packet Loop Avoidance

   An ELP that is first used by an ITR must be inspected for encoding
   loops. If any RLOC appears twice in the ELP, it MUST not be used.

   Since it is expected that multiple mapping systems will be used,
   there can be a loop across ELPs when registered in different mapping
   systems. The TTL copying procedures for reencapsulating tunnels and
   recursive tunnels in [RFC6830] MUST be followed.

6. Service Chaining

   An ELP can be used to deploy services at each reencapsulation point
   in the network. One example is to implement a scrubber service when
   a destination EID is being DoS attacked. That is, when a DoS attack
   is recognized when the encapsulation path is between ITR and ETR, an
   ELP can be registered for a destination EID to the mapping database
   system. The ELP can include an RTR so the ITR can encapsulate
   packets to the RTR which will decapsulate and deliver packets to a
   scrubber service device. The scrubber could decide if the offending
   packets are dropped or allowed to be sent to the destination EID. In
   which case, the scurbber delivers packets back to the RTR which
   encapsulates to the ETR.

7. **RLOC Probing by RTRs**

   Since an RTR knows the next tunnel hop to encapsulate to, it can
   monitor the reachability of the next-hop RTR RLOC by doing RLOC-
   probing according to the procedures in [RFC6830]. When the RLOC is
   determined unreachable by the RLOC-probing mechanisms, the RTR can
   use another locator in the locator-set. That could be the final ETR,
   a RLOC of another RTR, or an ELP where it must search for itself and
   use the next RLOC in the ELP list to encapsulate to.

   RLOC-probing can also be used to measure delay on the path between
   RTRs and when it is desirable switch to another lower delay ELP.

8. **Interworking Considerations**

   [RFC6832] defines procedures for how non-LISP sites talk to LISP
   sites. The network elements defined in the Interworking
   specification, the proxy ITR (PITR) and proxy ETR (PETR) (as well as
   their multicast counterparts defined in [RFC6831]) can participate in
   LISP-TE. That is, a PITR and a PETR can appear in an ELP list and
   act as an RTR.

   Note when an RLOC appears in an ELP, it can be of any address-family.
   There can be a mix of IPv4 and IPv6 locators present in the same ELP.
   This can provide benefits where islands of one address-family or the
   other are supported and connectivity across them is necessary. For
   instance, an ELP can look like:

   (x4, a46, b64, y4, etr)

   Where an IPv4 ITR will encapsulate using an IPv4 RLOC 'x4' and 'x4'
   could reach an IPv4 RLOC 'a46', but RTR 'a46' encapsulates to an IPv6
   RLOC 'b64' when the network between them is IPv6-only. Then RTR
   'b64' encapsulates to IPv4 RLOC 'y4' if the network between them is
   dual-stack.

   Note that RTRs can be used for NAT-traversal scenarios [LISP-NATT] as
   well to reduce the state in both an xTR that resides behind a NAT and
   the state the NAT needs to maintain. In this case, the xTR only
   needs a default map-cache entry pointing to the RTR for outbound
   traffic and all remote ITRs can reach EIDs through the xTR behind a
   NAT via a single RTR (or a small set RTRs for redundancy).

   RTRs have some scaling features to reduce the number of locator-set
   changes, the amount of state, and control packet overhead:

   o When ITRs and PITRs are using a small set of RTRs for
      encapsulating to "orders of magnitude" more EID-prefixes, the
      probability of locator-set changes are limited to the RTR RLOC
      changes versus the RLOC changes for the ETRs associated with the
      EID-prefixes if the ITRs and PITRs were directly encapsulating to
      the ETRs. This comes at an expense in packet stretch, but
      depending on RTR placement, this expense can be mitigated.

   o When RTRs are on-path between many pairwise EID flows, ITRs and
      PITRs can store a small number of coarse EID-prefixes.

   o RTRs can be used to help scale RLOC-probing. Instead of ITRs
      RLOC-probing all ETRs for each destination site it has cached, the
      ITRs can probe a smaller set of RTRs which in turn, probe the
      destination sites.

9. Multicast Considerations

   ELPs have application in multicast environments. Just like RTRs can
   be used to provide connectivity across different address family
   islands, RTRs can help concatenate a multicast region of the network
   to one that does not support native multicast.

   Note there are various combinations of connectivity that can be
   accomplished with the deployment of RTRs and ELPs:

   o Providing multicast forwarding between IPv4-only-unicast regions
      and IPv4-multicast regions.

   o Providing multicast forwarding between IPv6-only-unicast regions
      and IPv6-multicast regions.

   o Providing multicast forwarding between IPv4-only-unicast regions
      and IPv6-multicast regions.

   o Providing multicast forwarding between IPv6-only-unicast regions
      and IPv4-multicast regions.

   o Providing multicast forwarding between IPv4-multicast regions and
      IPv6-multicast regions.

   An ITR or PITR can do a (S-EID,G) lookup into the mapping database.
   What can be returned is a typical locator-set that could be made up
   of the various RLOC addresses:

   Multicast EID key: (seid, G)
   Locator-set: ETR-A: priority 1, weight 25
                       ETR-B: priority 1, weight 25
                       g1:   priority 1, weight 25
                       g2:   priority 1, weight 25

         An entry for host 'seid' sending to application group 'G'

   The locator-set above can be used as a replication list. That is
   some RLOCs listed can be unicast RLOCs and some can be delivery group
   RLOCs. A unicast RLOC in this case is used to encapsulate a
   multicast packet originated by a multicast source EID into a unicast
   packet for unicast delivery on the underlying network. ETR-A could
   be a IPv4 unicast RLOC address and ETR-B could be a IPv6 unicast RLOC
   address.

   A delivery group address is used when a multicast packet originated
   by a multicast source EID is encapsulated in a multicast packet for

multicast delivery on the underlying network. Group address 'g1'
could be a IPv4 delivery group RLOC and group address 'g2' could be
an IPv6 delivery group RLOC.

Flexibility for these various types of connectivity combinations can
be achieved and provided by the mapping database system. And the RTR
placement allows the connectivity to occur where the differences in
network functionality are located.

Extending this concept by allowing ELPs in locator-sets, one could
have this locator-set registered in the mapping database for (seid,
G). For example:

Multicast EID key: (seid, G)
Locator-set: (x, y, ETR-A):   priority 1, weight 50
                    (a, g, b, ETR-B): priority 1, weight 50

                    Using ELPs for multicast flows

In the above situation, an ITR would encapsulate a multicast packet
originated by a multicast source EID to the RTR with unicast RLOC
'x'. Then RTR 'x' would decapsulate and unicast encapsulate to RTR
'y' ('x' or 'y' could be either IPv4 or IPv6 unicast RLOCs), which
would decapsulate and unicast encapsulate to the final RLOC 'ETR-A'.
The ETR 'ETR-A' would decapsulate and deliver the multicast packet
natively to all the receivers joined to application group 'G' inside
the LISP site.

Let's look at the ITR using the ELP (a, g, b, ETR-B). Here the
encapsulation path would be the ITR unicast encapsulates to unicast
RLOC 'a'. RTR 'a' multicast encapsulates to delivery group 'g'. The
packet gets to all ETRs that have joined delivery group 'g' so they
can deliver the multicast packet to joined receivers of application
group 'G' in their sites. RTR 'b' is also joined to delivery group
'g'. Since it is in the ELP, it will be the only RTR that unicast
encapsulates the multicast packet to ETR 'ETR-B'. Lastly, 'ETR-B'
decapsulates and delivers the multicast packet to joined receivers to
application group 'G' in its LISP site.

As one can see there are all sorts of opportunities to provide
multicast connectivity across a network with non-congruent support
for multicast and different address-families. One can also see how
using the mapping database can allow flexible forms of delivery
policy, rerouting, and congestion control management in multicast
environments.

10. Security Considerations

   When an RTR receives a LISP encapsulated packet, it can look at the
   outer source address to verify that RLOC is the one listed as the
   previous hop in the ELP list. If the outer source RLOC address
   appears before the RLOC which matches the outer destination RLOC
   address, the decapsulating RTR (or ETR if last hop), MAY choose to
   drop the packet.

11. IANA Considerations

   At this time there are no requests for IANA.

12. References

12.1. Normative References

[RFC0791]   Postel, J., "Internet Protocol", STD 5, RFC 791,
            September 1981.

[RFC1034]   Mockapetris, P., "Domain names - concepts and facilities",
            STD 13, RFC 1034, November 1987.

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460]   Deering, S. and R. Hinden, "Internet Protocol, Version 6
            (IPv6) Specification", RFC 2460, December 1998.

[RFC3261]   Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
            A., Peterson, J., Sparks, R., Handley, M., and E.
            Schooler, "SIP: Session Initiation Protocol", RFC 3261,
            June 2002.

[RFC6830]   Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The
            Locator/ID Separation Protocol (LISP)", RFC 6830,
            January 2013.

[RFC6831]   Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The
            Locator/ID Separation Protocol (LISP) for Multicast
            Environments", RFC 6831, January 2013.

[RFC6832]   Lewis, D., Meyer, D., Farinacci, D., and V. Fuller,
            "Interworking between Locator/ID Separation Protocol
            (LISP) and Non-LISP Sites", RFC 6832, January 2013.

[RFC6833]   Fuller, V. and D. Farinacci, "Locator/ID Separation
            Protocol (LISP) Map-Server Interface", RFC 6833,
            January 2013.

12.2. Informative References

[LISP-LCAF]
            Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical
            Address Format", draft-ietf-lisp-lcaf-02.txt (work in
            progress).

[LISP-NATT]
            Ermagan, V., Farinacci, D., Lewis, D., Skriver, J., Maino,
            F., and C. White, "NAT traversal for LISP",
            draft-ermagan-lisp-nat-traversal-03.txt (work in

progress).

Appendix A. Acknowledgments

The authors would like to thank the following people for their ideas and comments. They are Albert Cabellos, Khalid Raza, and Vina Ermagan, and Gregg Schudel.

Appendix B. Document Change Log

B.1. Changes to draft-farinacci-lisp-te-03.txt

   o Update LISP references to their RFC pointers and document timer.

B.2. Changes to draft-farinacci-lisp-te-02.txt

   o Update references and document timer.

B.3. Changes to draft-farinacci-lisp-te-01.txt

   o Posted July 2012.

   o Add the Lookup bit to allow an ELP to be a list of encapsulation
      and/or mapping database lookup addresses.

   o Indicate that ELPs can be used for service chaining.

   o Add text to indicate that Map-Notify messages can be sent to new
      RTRs in a ELP so their map-caches can be pre-populated to avoid
      mapping database lookup packet loss.

   o Fixes to editorial comments from Gregg.

B.4. Changes to draft-farinacci-lisp-te-00.txt

   o Initial draft posted March 2012.

Authors' Addresses

    Dino Farinacci
    lispers.net
    San Jose, California
    USA

    Phone: 408-718-2001
    Email: farinacci@gmail.com

    Parantap Lahiri
    Juniper Networks
    Sunnyvale, CA
    USA

    Email: parantap.lahiri@gmail.com

    Michael Kowal
    cisco Systems
    111 Wood Avenue South
    ISELIN, NJ
    USA

    Email: mikowal@cisco.com